A Generic Approach for Obtaining Higher Entertainment in Predator/Prey Games

Georgios N. Yannakakis and John Hallam

Abstract— **This paper constitutes a sequel to our previous work focused on investigating cooperative behaviors, adaptive learning and on-line interaction towards the generation of entertainment in computer games. A human-verified metric of interest (i.e. player entertainment) of predator/prey games and a neuro-evolution on-line learning (i.e. during play) approach are used to serve this purpose. Experiments presented here demonstrate the generality of the proposed approach, in its ability to overcome the difficulties of learning in real-time and generate interesting predator/prey games, over game type, complexity, topology, initial opponent behavior and player type.**

Index Terms— **computer games, entertainment modeling, machine learning, cooperation, on-line interaction**

I. Introduction

IN our previous work [1], we defined criteria that contribute to the satisfaction for the player which map to characteristics of the opponent behavior in computer games. According to our hypothesis, the player-opponent interaction — rather than the audiovisual features, the context or the genre of the game — is the property that primarily contributes the majority of the quality features of entertainment in a computer game [1]. Based on this fundamental assumption, we introduced a metric for measuring the real-time entertainment value of predator/prey games which was established as an efficient and reliable entertainment metric by the approval of human judgement [2]. More specifically, Yannakakis and Hallam [2] reveal a high correlation between the human notion of interest and the proposed interest metric in one of the most representative test-beds of this computer game genre, that is Pac-Man.

According to our second hypothesis entertainment is generated when adaptive learning procedures occur in real-time. That allows for opponents to learn while playing against the player and adapt with regards to his/her strategy. Using the Pac-Man game as a test-bed [1], [3] and focusing on the non-player characters' (NPC's) behavior, a robust on-line neuro-evolution learning mechanism was presented which was capable of increasing the game's interest value as well as keeping that interest value at high levels while the game was being played. Moreover, it was found that when the game's entertainment value is at high levels, cooperative action among the opponents is present which furthermore contributes to player satisfaction.

G. N. Yannakakis is with the School of Informatics, Institute of Perception, Action and Behaviour, The University of Edinburgh, EH8 9LE, UK (e-mail: g.yannakakis@sms.ed.ac.uk).

J. Hallam is with the Mærsk Mc-Kinney Møller Institute for Production Technology, University of Southern Denmark, Campusvej 55, DK-5230 Odense M, Denmark (e-mail: john@mip.sdu.dk).

The approach embeds a replacement method of the worst-fit individuals while playing the game. This mechanism demonstrated high robustness and adaptability to changing hand-crafted player strategies in a relatively simple playing stage. Additional experiments in a predator/prey game of a more abstract design called 'Dead End' [4] displayed the effectiveness of the proposed methodology in dissimilar games of the same genre and expanded the applicability of the method.

In the work presented here, we attempt to test the proposed on-line learning mechanism's ability to generate interesting games over a number of computer game dimensions. These dimensions include (a) the (high-level) concept of the game; (b) the environment of the game and particularly the complexity of the game world (i.e. stage) and its topological features; (c) the opponents' behavior when the game starts and (d) the player's gaming skills. Experiments presented here demonstrate the generality of the methodology with respect to the aforementioned dimensions and verify our hypothesis that the proposed on-line learning mechanism constitutes a generic tool for obtaining predator/prey games of high entertainment independently of game type, game complexity and topology, initial opponent behavior and player.

We have chosen predator/prey games as the initial genre of our game research since, given our aims, they provide us with unique properties. In such games we can deliberately abstract the environment and concentrate on the characters' behavior. The examined behavior is cooperative since cooperation is a prerequisite for effective hunting behaviors. Furthermore, we are able to easily control a learning process through on-line interaction. In other words, predator/prey games offer a well-suited arena for initial steps in studying cooperative behaviors generated by interactive on-line learning mechanisms. Other genres of game (e.g. first person shooters) offer similar properties; however predator/prey games are chosen for their simplicity as far as their development and design are concerned.

II. Entertainment Metrics

The current state-of-the-art in machine learning in games is mainly focused on generating human-like [5], [6] and intelligent [7], [8] characters. Even though complex opponent behaviors emerge through various learning techniques, there is no further analysis of whether these behaviors contribute to the satisfaction of the player. In other words, researchers hypothesize — for instance by observing the vast number of multi-player on-line games played daily on the web — that by generating human-like opponents [9] they enable the player to gain more satisfaction from the game. According to Taatgen et al. [10] believability of computer game opponents, which are generated through cognitive models, is strongly correlated with enjoyable games. These hypotheses might be true up to a point; however, since a notion of interest has not been explicitly defined, there is no evidence that a specific opponent behavior generates enjoyable games. This statement is the core of Iida's work on entertainment metrics for variants of chess games [11].

Inspired by Iida's metric of entertainment and based on Malone's theoretical approach on entertainment factors, namely challenge, curiosity and fantasy, for simple arcade games [12] and Csikszentmihalyi's theory of *flow* [13], Yannakakis and Hallam [1] introduced an efficient metric of real-time entertainment for computer games. This metric has been used for variants of predator/prey games and has been successfully cross-validated with human notion of entertainment [2].

III. Evolutionary Learning in Real-Time

Evolutionary computation is not very well studied and explored in the area of computer games (unlike card or board games simulated in computers — e.g. chess). In particular, the primary reason against its use for learning while playing (on-line) is its slow convergence and that undesired/unpredictable behaviors may emerge. However, real-time adaptation which intelligent opponents exhibit is the main feature that motivates research on on-line evolutionary learning. Very recently, successful on-line neuro-evolution applications [14], [3] demonstrate the feasibility of the method through more efficient learning procedures and careful representation design.

Evolutionary learning of neural controlled NPCs is the core of the work of Yannakakis et al. [1], [3], [4], [15]. Among their contributions, a robust and highly adaptive on-line evolutionary learning mechanism is presented. The predator/prey genre of games is this approach's test-bed. In the same research direction, Stanley et al. [14] are applying neuro-evolution techniques for the emergence of adaptive behaviors (e.g. capture-the-flag and wall-avoidance) in the 'NERO' training game in real-time. For this game, the NeuroEvolution Augmenting Topologies (NEAT) method [16] has been used to evolve large populations of NPCs through an on-line replacement mechanism of the worst-fit NPCs. In NERO, both the game genre (training) and the number (fifty) of NPCs contribute to the efficiency and the convergence time of the mechanism. Both leave space for unpredictable and/or unwanted emergent opponent behaviors to be accepted and/or ignored by the player. On the other hand, in [3], on-line evolutionary learning is successfully applied in a computer game of four opponents where no concession is made by the player for unrealistic behaviors.

IV. Learning in Real-Time Challenges

We can distinguish the challenges we will come across in this paper into the ones generated by the game design per se and the ones generated by learning on-line in computer games. The on-line learning challenges are introduced in this section. We make this distinction here since the difficulties arising from learning in real-time will be faced globally throughout the paper.

The drawbacks we have to overcome when designing a learning (neuro-evolution) mechanism for real-time opponent adaptation are as follows:

- **Real-time performance**. An on-line learning approach should perform fast in real-time since only a single CPU is available for the majority of computer games [17]. Note also, that most commercial computer games use over 90% of the CPU for their graphics engines alone.
- **Realism**. On-line neuro-evolution provides the potential for adaptive opponents but it may also generate unrealistic opponent behaviors. In most computer games the player interacts with a very small number of opponents in the majority of gameplay instances. Unrealistic behaviors are, therefore, easily noticeable and may lead to low entertainment for the player.
- **Fast adaptation**. A successful on-line learning approach should adapt quickly to changes in the player's strategy. Otherwise, the game becomes boring. This constitutes a big challenge for the design of the on-line learning mechanism given the computational power resources, the realistic behavior condition presented before and the small number of opponents that normally interact with the player in computer games.

The predator/prey games used in this paper face all aforementioned challenges of on-line learning design since experiments are held in a single 1GHz CPU and the number of opponents is less than or equal to five.

V. Real-Time Interest

A complete methodology for defining a generic measure of interest for predator/prey games is introduced in [2]. The approach proposed portrays the criteria that make predator/prey games interesting, quantifies and combine all these criteria in a mathematical formula and uses a test-bed game in order to have this formulation of interest cross-validated against the interest the game produces when it is played by human players. Results from this work showed that the perceived human entertainment is highly correlated (correlation coefficient equals 0.4444, p-value equals $1.17 \cdot 10^{-8}$) with the interest metric proposed and demonstrated the reliability of the metric. An outline of the methodology followed in [2] in order to obtain the interest metric is presented here.

The methodology is primarily focused on the opponents' behavior contributions to the interest value of the game because, we believe, the computer-guided opponent character can contribute to the majority of qualitative features that make a game interesting. The player, however, may contribute to its entertainment through its interaction with the opponents of the game and, therefore, it is implicitly included in the interest formulation presented here.

By observing the opponents' behavior of various predator/prey games we attempted to empirically extract the features that may generate entertainment for the player. These features where experimentally cross-validated against various opponents of different strategies and redefined when appropriate.

Hence, by following the theoretical principles of Malone's intrinsic factors for engaging gameplay [12] and the basic concepts of *flow* (the mental state in which players are so involved in something that nothing else matters) [13], the criteria that collectively define interest on any predator/prey game are:

1) *When the game is neither too hard nor too easy*. In other words, the game is interesting when opponents manage to kill the player sometimes but not always. In that sense, highly-effective opponent behaviors are not interesting behaviors and *vice versa*.
2) *When there is diversity in opponents' behavior over the games*. That is, when the non-player characters are able to find different ways of hunting and killing the player in each game so that their strategy is less predictable.
3) *When opponents' behavior is aggressive rather than static*. That is, predators that move constantly all over the game world and cover it uniformly. This behavior gives the player the impression of an intelligent strategic opponents' plan which increases the player's curiosity ("*what will happen next?*"). According to Malone [12], curiosity is one of the three main features that generate entertainment in computer games.

The metrics for the three criteria, as defined in [2], are estimated by T (appropriate level of challenge metric; based on the difference between maximum player's lifetime t_{max} and average player's lifetime $E\{t_k\}$ over N games — N is 50 in this paper), S (behavior diversity metric; based on standard deviation of player's lifetime over N games) and $E\{H_n\}$ (spatial diversity metric; based on stage grid-cell visit average entropy of the opponents over N games) respectively. All three metrics are combined linearly thus

$$I = \frac{\gamma T + \delta S + \epsilon E\{H_n\}}{\gamma + \delta + \epsilon} \tag{1}$$

where I is the interest value of the predator/prey game; γ, δ and ϵ are criterion weight parameters.

The above-mentioned interest metric can be applied effectively to any predator/prey computer game because it is based on generic features of this category of games. These features include the time required to kill the prey as well as the predators' distribution across the game field. We therefore believe that (1) — or a similar measure of the same concepts — constitutes a generic interest approximation of predator/prey computer games. Moreover, given the two first interest criteria previously defined, the approach generalizes to all computer games. Indeed, no player likes any computer game that is too hard or too easy to play and, furthermore, any player would enjoy diversity throughout the play of any game. The third interest criterion is applicable to games where spatial diversity is important which, apart from predator/prey games, may also include action, strategy and team sports games according to the computer game genre classification of Laird and van Lent [5].

VI. The Pac-Man game

The first test-bed studied is a modified version of the original Pac-Man computer game released by Namco. The player's (*PacMan*'s) goal is to eat all the pellets appearing in a maze-shaped stage while avoiding being killed by the four *Ghosts*. The game is over when either all pellets in the stage are eaten by *PacMan*, *Ghosts* manage to kill *PacMan* or a predetermined number of simulation steps is reached without any of the above occurring. In that case, the game restarts from the same initial positions for all five characters.

The game is investigated from the opponents' viewpoint and more specifically how the *Ghosts'* emergent adaptive behaviors can collectively contribute to the player's entertainment. The game field (i.e. stage) consists of corridors and walls whereas both the stage's dimensions and its maze structure are predefined. For the experiments presented in this paper we use a 19×29 grid maze-stage where corridors are one grid-cell wide.

A. PacMan

In [1], three fixed *Ghost*-avoidance and pellet-eating strategies for the *PacMan* player, differing in complexity and effectiveness are presented. Each strategy is based on decision making applying a cost or probability approximation to the player's 4 neighbor cells (i.e. up, down, left and right). We present them briefly in this paper.

- Cost-Based (CB) *PacMan*: moves towards a cost minimization path that produces effective *Ghost*-avoidance and (to a lesser degree) pellet-eating behaviors but only in the local neighbor cell area.
- Rule-Based (RB) *PacMan*: is a CB *PacMan* plus an additional rule for more effective and global pellet-eating behavior.
- Advanced (ADV) *PacMan*: the ADV moving strategy generates a more global *Ghost*-avoidance behavior built upon the RB *PacMan*'s good pellet-eating strategy.

B. Ghosts

A multi-layered fully connected feedforward neural controller, where the sigmoid function is employed at each neuron, manages the *Ghosts'* motion. Using their sensors, *Ghosts* inspect the environment from their own point of view and decide their next action. Each *Ghost*'s perceived input consists of the relative coordinates of *PacMan* and the closest *Ghost*. We deliberately exclude from consideration any global sensing, e.g. information about the dispersion of the *Ghosts* as a whole, because we are interested specifically in the minimal sensing scenario. We make this choice by following principles of the *animat* approach [18] which provides the ground for realistic NPC behaviors in computer games [19] and the potential of scaling up to more complex game environments. The neural network's output is a four-dimensional vector with respective values from 0 to 1 that represents the *Ghost*'s four movement options (up, down, left and right respectively). Each *Ghost* moves towards the available — unobstructed by walls — direction represented by the highest output value. Available movements include the *Ghost*'s previous cell position.

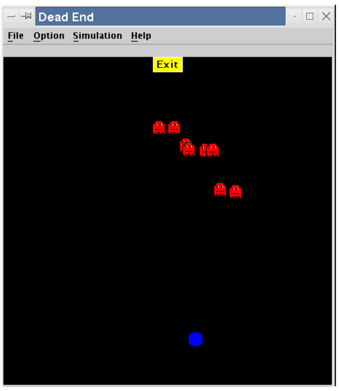


Fig. 1. A snapshot of the Dead End game.

VII. The Dead End Game

Conceptually, the primary dissimilarities between Pac-Man and Dead End (the game presented in this section) are found in:

- The player's objectives.
- The type of opponent motion; in both games the movement directions are limited to up, down, left or right but while the *Ghosts*' magnitude of motion is measured in grid cells the Dead End opponents' magnitude of motion is measured in pixels. Moreover, the ratio of the player's over the opponents' maximum speed is 4/3 and 2 for the games of Dead End and Pac-Man respectively.
- The absence of objects-walls in Dead End.

The Dead End game field (i.e. stage) is a two-dimensional square world that contains a white rectangular area named "Exit" (see Fig. 1) at the top. For the experiments presented in this paper we use a 215×215 pixel stage (see Fig. 1). The characters visualized in the Dead End game (as illustrated in Fig. 1) are a dark grey circle of radius 10 pixels representing the player, named '*Cat*', and a number of light grey square (of dimension 20 pixels) ghost-like characters representing the opponents, named '*Dogs*'. *Cat* and *Dogs* are initially placed in the game field so that there is a suitably large distance between them. The aim of the *Cat*, starting from a randomly chosen position at the bottom of the stage, is to reach the Exit by avoiding the *Dogs* or to survive for a predetermined large period of time of 50 simulation steps. On the other hand, *Dogs* are aiming to defend the Exit and/or catch the *Cat* within that period of time. The game's fundamental concepts are inspired by previous work of Yannakakis et al. [20] while the first use of the game as a test-bed for experiments on emergent cooperative opponent behaviors is introduced in [21].

At each simulation step, the *Cat* moves at four thirds the *Dogs*' maximum speed, and since there are no dead ends, it is impossible for a single *Dog* to complete the task of killing it. Since the player moves faster than its opponents, the only effective way to kill the *Cat* is for a group of *Dogs* to hunt cooperatively. When the game is over, then a new game starts from the same initial positions for the *Dogs* but from a different, randomly chosen, position at the bottom of the stage for the *Cat*.

A. Cat

In [15], we introduced three hand-crafted *Dog*-avoidance and/or Exit-achieving strategies for the *Cat*, differing in complexity and effectiveness. These are briefly described here as follows:

- Randomly-Moving (RM) *Cat*: takes a movement decision by selecting a uniformly distributed random picked direction at each simulation step of the game. The probability of selecting the direction towards the Exit linearly increases over the simulation steps by 0.2% per step.
- Exit-Achieving (EA) *Cat*: moves directly towards the Exit. Its strategy is based on moving so as to reduce the greatest of its relative coordinates from the Exit.
- Potential Field-Based (PFB) *Cat*: this constitutes the most efficient *Dog*-avoidance and Exit-achieving strategy of the three different fixed-strategy types of player. A discrete Artificial Potential Field (APF) [22], specially designed for the Dead End game, controls the PFB player's motion. The overall APF causes a force to act on the *Cat* which guides it along a *Dog*-avoidance Exit-achievement path. For a more detailed presentation of the PFB *Cat*, see [15] (appears as 'CB player').

B. Dogs

Dogs' motion — like the *Ghosts'* motion presented in Section VI-B — is managed by a feedforward neural controller where the hyperbolic tangent sigmoid function is employed at each neuron. The neural network's input array determines the *Dog*'s perceived information and consists of the relative coordinates of (a) the *Cat*, (b) the closest *Dog* and (c) the Exit. A *Dog*'s input includes information for only one neighbor *Dog* as this constitutes the minimal information for emerging cooperative teamwork. We are primarily interested in the minimal sensing scenario and therefore global information is deliberately not considered. The neural network's output is a two-dimensional vector with respective values from -1 to 1 that represent movement in the X (positive sign corresponds to up, negative to down) and Y (positive sign corresponds to left, negative to right) directions. Each *Dog* moves towards the direction represented by the highest absolute output value at the magnitude determined by this value multiplied by the *Dog*'s maximum speed (i.e. 20 pixels/simulation step).

VIII. Fixed Strategy Opponents

Apart from the neural controlled opponents, two additional non-evolving strategies have been tested for controlling the opponent's motion for both games. These strategies are used as baseline behaviors for comparison with any neural controller emerged behavior.

- Random (R): opponents that randomly decide their next available movement. Available movements have equal probabilities of being picked.
- Followers (F): opponents designed to follow the player constantly. Their strategy is based on moving (at maximum speed) so as to reduce the greatest of their relative coordinates from the player.

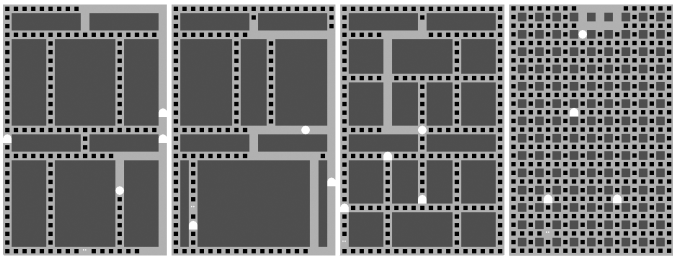


Fig. 2. The 4 different stages of the Pac-Man game. Complexity increases from left to right: Easy (A and B), Normal and Hard.

IX. Game Complexity

In order to distinguish between environments of different complexity within the same predator/prey game, we require an appropriate measure to quantify this feature. Because of the different features of the two games examined, two different measures are proposed. For the Pac-Man game this measure, C, is

$$C = 1/E\{L\} \tag{2}$$

where $E\{L\}$ is the average corridor length of the stage (see also [3]).

According to (2), complexity is inversely proportional to the average corridor length of the stage. That is, the longer the average corridor length, the easier for the opponents to block the player and, therefore, the less complex the stage.

Fig. 2 illustrates the four different stages used for the experiments presented here. Complexity measure values for the Easy A, Easy B, Normal and Hard stages are 0.16, 0.16, 0.22 and 0.98 respectively. Furthermore, given that a) blocks of walls should be included b) corridors should be one grid-square wide and c) dead ends should be absent, Hard stage is the most complex Pac-Man stage of that size for the *Ghosts* to play.

For the Dead End game, complexity is measured by the number of opponents since conceptually the game environment is free of wall blocks[1]. That is, the more *Dogs* in the game the easier their hunting task. Five, four and three *Dog* game environments have been used for the experiments presented in this paper.

A. Topology

Stages of the same complexity, measured by (2), may differ in topology (i.e. layout of blocks on the stage). Thus the case of Easy A and Easy B stages (see Fig. 2) have the same complexity value but are topologically different. The choice of these two Pac-Man stages is made so as to examine the on-line learning mechanism's ability to generate interesting opponents in equally complex stages of different topology — that is, to investigate the topology's impact on the generation of interesting games.

[1]The number of *Ghosts* can give an indication for the complexity of the Pac-Man game as well; however, by following the original version of the game containing four *Ghosts* we keep this variable constant.

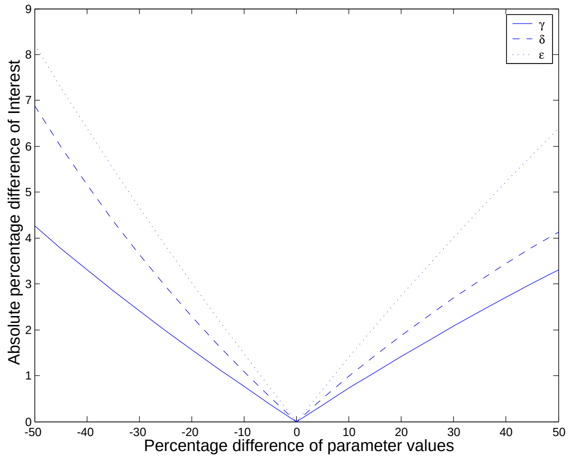
X. Interest Parameter Values for the Games

In this section we present the procedures followed to obtain the appropriate parameter values of the interest estimate (1) for both games tested. As previously defined, t_{max} is the maximum evaluation period of play, or else the maximum lifetime of the player. For Pac-Man this number corresponds to the minimum simulation period required by the RB *PacMan* (best pellet-eater) to clear the stage of pellets. In the experiments presented here t_{max} is 300 for the Easy stage, 320 for the Normal stage and 466 for the Hard stage. In the game of Dead End t_{max} determines the maximum game length (i.e. until a *win* or a *kill* is occurred) recorded against any opponent *Dogs* and it is 50 simulation steps for the RM and PFB *Cat* and 10 simulation steps for the EA *Cat*.

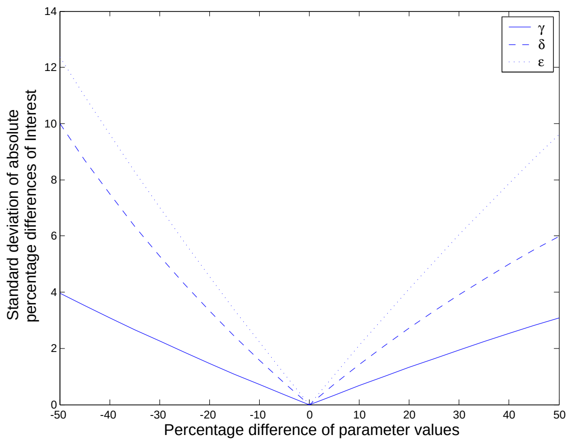
In order to obtain values for the interest formula weighting parameters γ, δ and ϵ we select empirical values based on the specific game. For Pac-Man we select $\gamma = 1, \delta = 2$ and $\epsilon = 3$; a sensitivity analysis for these values, in [2], showed that the interest value does not change significantly even when a 40% change in the parameter value occurs. For Dead End, diversity in game play is of greatest entertainment value. The average game length is much smaller when compared to Pac-Man and, therefore, we believe that generating diverse behaviors within this period of play should be weighted more than the challenge (T) and the spatial diversion ($E\{H_n\}$) criteria. Given the above-mentioned statements and by adjusting these three parameters so that the interest value escalates as the opponent behavior changes from Random to Follower, we come up with $\gamma = 1, \delta = 2$ and $\epsilon = 1$.

Since the interest value changes monotonically with respect to each of the three criterion values T, S, $E\{H_n\}$, sensitivity analysis is conducted on the above interest metric parameters aiming to portray the relation between these parameters as well as their weighting degree in the interest formula. We therefore proceed by seeking opponent behaviors that generate ten different T, S and $E\{H_n\}$ values, equally spread in the [0,1] interval. Given these thirty values as input, γ, δ and ϵ parameters are systematically changed one at a time so that their percentage difference lies in the interval $[-50\%, 50\%]$. We believe that fifty is a large enough percentage difference to demonstrate potential significant impact on the observed value. Each time a parameter change occurs, the absolute percentage difference of the game's interest value is computed. The function between the absolute percentage differences of the interest value and the percentage differences of the interest weighting parameters is illustrated in Fig. 3(a) and Fig. 3(b).

As seen from Fig. 3(a), δ and ϵ demonstrate significant differences (i.e. greater than 5%) in I when decreased by 40% and 30% respectively. The ϵ parameter also significantly changes the I value when it is increased by 35%. Finally, for γ no significant change in I is observed even when changed by up to 50%. Results obtained illustrate that exploring the impact of combinations of parameter changes (comprehensive approach) would not be necessary since significant changes occur only in the rather extreme cases of $\epsilon < 0.7$, $\epsilon > 1.35$ and $\delta < 1.2$. In addition, such an approach would increase the size of different interest value calculations up to 10^4.



(a) Average



(b) Standard deviation

Fig. 3. Average and standard deviation of absolute percentage differences of I over ten runs for each weighting parameter.

XI. On-Line Learning

On-line learning is based on the idea of opponents that learn while they are playing against the player: *Ghosts* or *Dogs* that adapt to any player's behavior and learn from its strategy instead of being characters of predictable and, to a degree, uninteresting behaviors. Furthermore, this approach's additional objective is to keep the game's interest value at high levels as long as it is being played.

Beginning from any initial group of homogeneous off-line trained (see [1] and [15] for more details on Pac-Man and Dead End off-line training mechanisms respectively) opponents, the OLL mechanism transforms them into a group of heterogeneous characters that are conceptually more interesting to play against. In OLL, an opponent trained off-line is cloned a number of times equal to the number of opponents playing the game, and its clones are placed in the game field to play against a selected fixed player type in a selected stage. Then, at each generation:

Step 1 Each opponent is evaluated every e_p simulation steps via (3), while the game is played — e_p is 25 simulation steps in this paper.

$$f_{OLL} = \sum_{i=1}^{e_p} \left\{ D_{P,i} - D'_{P,i} \right\} \qquad (3)$$

where

$$D_{P,i} = |x_o^{i+1} - x_p^i| + |y_o^{i+1} - y_p^i| \qquad (4)$$
$$D'_{P,i} = |x_o^i - x_p^i| + |y_o^i - y_p^i| \qquad (5)$$

and (x_o^i, y_o^i), (x_p^i, y_p^i) are respectively the cartesian coordinates of the opponent and the player at the i^{th} simulation step. This fitness function promotes opponents that move towards the player within an evaluation period of e_p simulation steps.

Step 2 A pure elitism selection method is used where only the fittest solution is able to breed. The fittest parent clones an offspring with a probability p_c that is inversely proportional to the normalized predators cell visit entropy (i.e. $p_c = 1 - H_n$). If there is no cloning, then go back to Step 1, else continue to Step 3.

Step 3 Mutation occurs in each gene (connection weight) of each offspring's genome with a probability p_m (e.g. 0.02). For the Dead End and the Pac-Man game, a uniform and a gaussian random distribution is respectively used to define the mutated value of the connection weight. Thus, for Pac-Man, the mutated value is obtained from (6),

$$w_m = \mathcal{N}(w, 1 - H_n) \qquad (6)$$

where w_m is the mutated connection weight value and w is the connection weight value to be mutated.

Step 4 The mutated offspring is evaluated briefly via (3) in off-line mode, that is, by replacing the least-fit member of the population and playing an off-line (i.e. no visualization of the actions) short game of e_p simulation steps. The fitness values of the mutated offspring and the least-fit opponent are compared and the better one is kept for the next generation. This pre-evaluation procedure for the mutated offspring attempts to minimize the probability of group behavior disruption by low-performance mutants. If the least-fit opponent is replaced, then the mutated offspring takes its position in the game field as well. This algorithm step is applied to the Pac-Man game only.

The algorithm is terminated when a predetermined number of games has been played or a number of generations has been achieved.

We mainly use short simulation periods (i.e. $e_p = 25$) to evaluate opponents in OLL, so as to accelerate the on-line evolutionary process. The same period is used for the evaluation of mutated offspring; this is based on two primary objectives: 1) to apply a fair comparison between the mutated offspring and the least-fit opponent (i.e. same evaluation period) and 2) to avoid undesired high computational effort while playing. However, the evaluation function (3) constitutes an approximation of the examined opponent's overall performance for large simulation periods. Keeping the right balance between computational effort and performance approximation is one of the key features of this approach and, therefore, we use minimal evaluation periods capable of achieving good performance estimation.

Note that, there are specific dissimilarities between the OLL approach applied to Pac-Man and Dead End. The purpose of these differences is the investigation of the pre-evaluation process' impact on the algorithm's performance. See Section XVI for a comprehensive discussion of this topic.

XII. PAC-MAN EXPERIMENTS

Off-line trained (OLT) emergent solutions (see [1] for more details of off-line training) are the OLL mechanisms' initial points in the search for more interesting games. OLT behaviors are classified into the following categories:

- Blocking (B): These are OLT *Ghosts* that tend to wait for *PacMan* to enter a specific area that is easy for them to block and kill. Their average normalized cell visit entropy value $E\{H_n\}$ lies between 0.55 and 0.65.
- Aggressive (A): These are OLT *Ghosts* that tend to follow *PacMan* all over the stage in order to kill it ($E\{H_n\} \geq 0.65$).
- Hybrid (H): These are OLT *Ghosts* that tend to behave as a Blocking-Aggressive hybrid which proves to be ineffective at killing *PacMan* ($E\{H_n\} < 0.55$).

A. OLL experiment

The OLL experiment is described as follows. a) Pick nine different emerged *Ghosts'* behaviors produced from off-line learning experiments — Blocking (B), Aggressive (A) and Hybrid (H) behaviors emerged by playing against each of 3 *PacMan* types — for each one of the three stages; b) starting from each OLT behavior, apply the OLL mechanism by playing against the same type of *PacMan* player and in the same stage the *Ghosts* have been trained in off-line. Initial behaviors for the Easy B stage are OLT behaviors emerged from the Easy A stage. This experiment intends to demonstrate the effect of the topology of a stage in the interest value of the game; c) calculate the interest value of the game every 100 games during each OLL attempt.

The interest value is calculated by letting the *Ghosts* play 100 non-evolution games in the same stage against the *PacMan* type they were playing against during OLL. In order to minimize the non-deterministic effect of the *PacMan*'s strategy on the *Ghost*'s performance and interest values as well as to draw a clear picture of these averages' distribution, we apply the following bootstrapping procedure. Using a uniform random distribution we pick 10 different 50-tuples out of the 100 above-mentioned games. These 10 samples of data, of 50 games each, are used to determine the games' average interest value as well as its confidence interval values. The outcome of the OLL experiment is presented in Fig. 4 and Fig. 5.

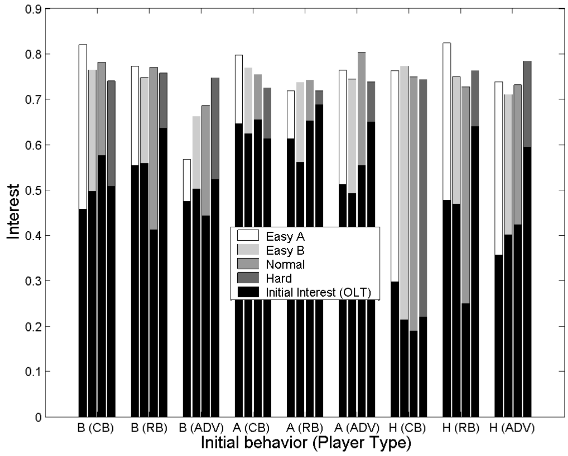


Fig. 4. On-line learning effect on the interest value of the game. Best interest values achieved from on-line learning on *Ghosts* trained off-line (B, A, H). Experiment Parameters: $t = 50$ simulation steps, $p_m = 0.02$, 5-hidden neurons controller.

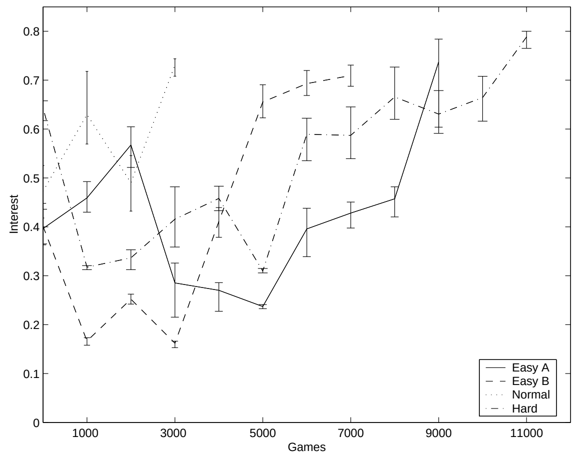


Fig. 5. On-line learning effect on interest value of ADV Hybrid initial behavior in all four stages. For reasons of computational effort, the OLL procedure is terminated when a game of high interest value ($I \geq 0.7$) is found.

Since there are 3 types of players, 3 initial OLT behaviors and 4 stages, the total number of different OLL experiments is 36. These experiments illustrate the overall picture of the mechanism's effectiveness over the complexity and the topology of the stage as well as the *PacMan* type and the initial behavior (see Fig. 4). Due to space considerations we present only 4 (see Fig. 5) out of the 36 experiments in detail here, where the evolution of the interest value over the OLL games (starting from the hybrid behavior emerged by playing against the ADV *Pac-Man* player) on each stage is illustrated.

As seen from Fig. 5, the OLL mechanism manages to find ways of increasing the interest value of the game regardless the stage complexity or topology. It is clear that the OLL approach constitutes a robust mechanism that, starting from suboptimal OLT *Ghosts*, generates interesting games (i.e. interesting *Ghosts*) in all 36 cases. It is worth mentioning that in 15 out of 36 different OLL attempts the best interest value is greater than the respective Follower's value (see Table I). Furthermore, in nearly all cases, the interest measure is kept

TABLE I

FIXED STRATEGY *Ghosts'* (R, F) INTEREST VALUES. VALUES ARE OBTAINED BY AVERAGING 10 SAMPLES OF 50 GAMES EACH.

			Play Against		
		Stage	CB	RB	ADV
Fixed Behaviors	R	Easy A	0.5862	0.6054	0.5201
		Easy B	0.5831	0.5607	0.4604
		Normal	0.5468	0.5865	0.5231
		Hard	0.3907	0.3906	0.3884
	F	Easy A	0.7846	0.7756	0.7759
		Easy B	0.7072	0.6958	0.6822
		Normal	0.7848	0.8016	0.7727
		Hard	0.7727	0.7548	0.7627

at the same level independently of stage complexity or — in the case of Easy A and B stages — stage topology. Given the confidence intervals (± 0.05 maximum, ± 0.03 on average) of the best interest values, it is revealed that the emergent interest value is not significantly different from stage to stage. Comprehensive results from the Easy A stage [1] present OLL experiments where OLT *Ghosts* play against all *PacMan* types instead of just the type that was used for their off-line training. The outcome of these experiments demonstrates the generality of the OLL over this additional variation.

However, a number in the scale of 10^3 constitutes an unrealistic number of games for a human player to play. On that basis, it is very unlikely for a human to play so many games in order to notice the game's interest value increasing. The reason for the OLL process being that slow is a matter of keeping the right balance between the process' speed and its 'smoothness' (by 'smoothness' we define the interest value's magnitude of change over the games). A solution to this problem is to consider the initial long period of disruption as an off-line learning procedure and start playing as soon as the game's interest value is increased.

XIII. DEAD END EXPERIMENTS

Exactly as in the Pac-Man game, some well-behaved *Dogs* are required initially from the OLL mechanism in its attempt to generate interesting Dead End games. Off-line trained emergent solutions (see [15] for more details on off-line training for the Dead End game) serve this purpose. OLT obtained behaviors are classified into the following two categories:

- Defensive (D): These are OLT *Dogs* that tend to flock close and around the Exit and wait for the *Cat* to approach in order to kill it. Their average normalized cell visit entropy value $E\{H_n\}$ is less than 0.7.
- Aggressive (A): These are OLT *Dogs* that tend to follow the *Cat* all over the stage in order to kill it ($E\{H_n\} \geq 0.7$).

Defending the Exit, as an emergent behavior, is much easier than hunting cooperatively and more effective when playing against the EA *Cat*. Thus, when off-line training occurs against the EA *Cat*, aggressive *Dog* behavior is not emerged because it constitutes a sub-optimal behavior.

A. OLL experiment

The OLL experiment for the Dead End game is described as follows. a) Pick five different emerged *Dogs'* behaviors produced from off-line learning experiments — Defensive (D) against each of the three *Cat* types and Aggressive (A) against the RM and the PFB *Cat* — for each one of the three game environments — containing eight, four and three *Dogs*; b) starting from each OLT behavior, apply the OLL mechanism by playing against each *Cat* type separately and in the same stage where off-line training occurred. This makes a total of fifteen different OLL attempts for each game environment. c) calculate the interest value (by following the bootstrapping procedure presented in Section XII-A) of the game every 100 generations during each OLL attempt. In contrary to the Pac-Man game, generations instead of games are picked as the algorithm's simulation time unit because of the large difference on the average game length between the EA *Cat* (i.e. $t_{max} = 10$) and the RM and PFB *Cat* (i.e. $t_{max} = 50$). For comparison purposes to the OLL experiments in the Pac-Man game, the expected value of the generations g per games G ratio $E\{g/G\}$ is calculated. This equals 0.943 for the RM and PFB *Cat* and 3.846 for the EA *Cat*.

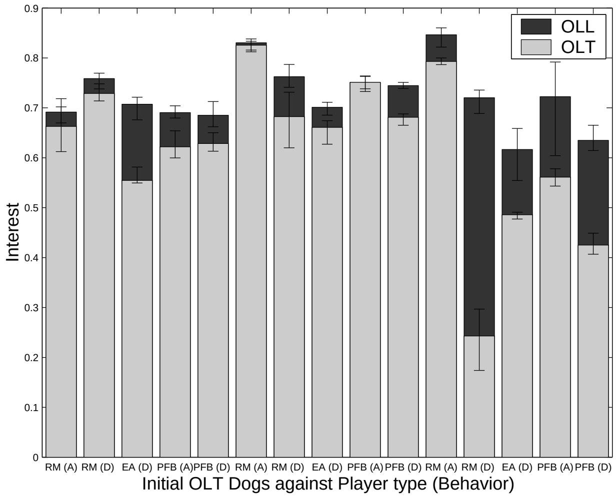
Given that there are three game environments explored, the total number of different OLL experiments is 45, which illustrate a complete picture of the mechanism's effectiveness over the complexity of the game, the *Cat* type and the initial behavior (see Fig. 6). Three of these experiments are presented in detail due to space considerations. More specifically, in Fig. 6(d), the evolution of the interest value over the OLL generations (starting from a D behavior trained against the RM *Cat* and playing against the PFB *Cat*) on each stage is illustrated.

TABLE II

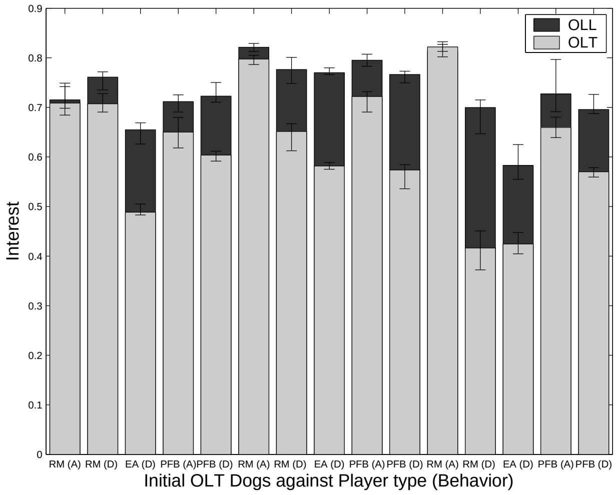
FIXED STRATEGY *Dogs'* (R, F) INTEREST VALUES. VALUES ARE OBTAINED BY AVERAGING 10 SAMPLES OF 50 GAMES EACH.

			Play Against		
		Environment	RM	EA	PFB
Fixed Behaviors	R	5 Dogs	0.491	0.498	0.425
		4 Dogs	0.614	0.436	0.156
		3 Dogs	0.561	0.312	0.236
	F	5 Dogs	0.607	0.778	0.783
		4 Dogs	0.684	0.791	0.768
		3 Dogs	0.624	0.797	0.772

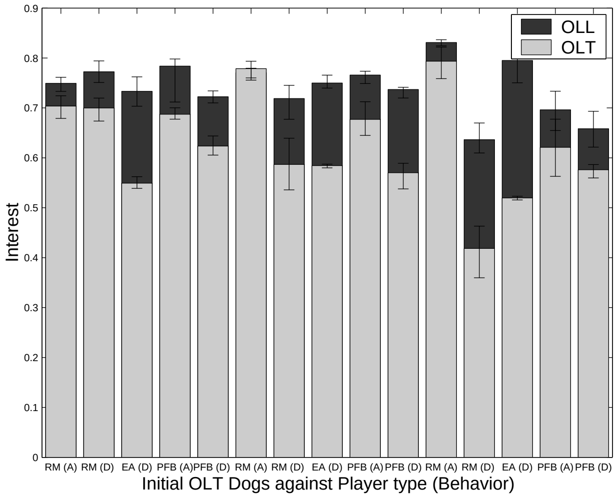
As seen from Fig. 6, OLL is successful at augmenting the interest value of the game regardless the stage complexity, stage topology, the *Cat* type and the initial behavior. On that basis, in the majority of the experiments, OLL is capable of producing games of higher than the initial interest value as well as maintaining that high interest value for a long period. More comprehensively, in 42 out of 45 OLL scenarios the interest value of the game is increased in fewer than 500 generations and in 37 cases this increase is statistically significant. Also, in 23 cases the best interest value achieved against a *Cat* type is greater than the respective interest value generated by the Followers (see Table II). Given the calculated



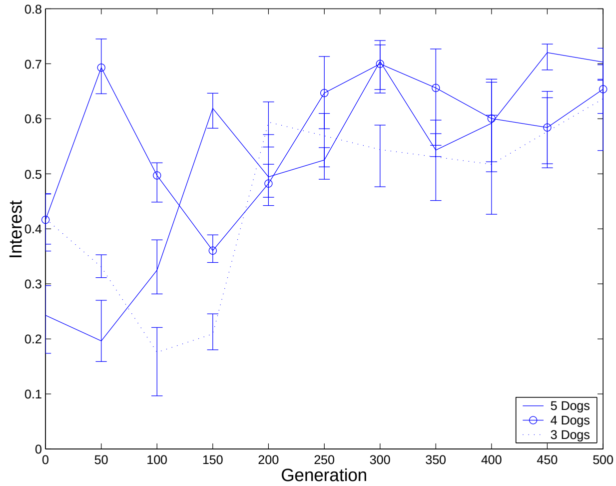
(a) 5 Dogs



(b) 4 Dogs



(c) 3 Dogs



(d) Initial Behavior: RM (D)

Fig. 6. (a), (b), (c): best interest values achieved from OLL in the three Dead End environments; (d) game interest value over the number of OLL generations — OLL against the PFB *Cat* starting from a D behavior OLT against the RM *Cat* in each environment Experiment Parameters: $e_p = 25$ simulation steps, $p_m = 0.02$, 5-hidden neurons controller.

confidence intervals of the interest value, in 17 of such cases this difference is significant.

XIV. How Does OLL Work?

The fitness function (3) rewards ‘aggressiveness’ (moving toward the player); however, the OLL generated opponents become eventually more interesting. Herein, we will attempt to explain theoretically the correlation between aggression and interest that initially appears to be rather unexpected.

The OLL algorithm promotes mutation of groups of opponents that exhibit low spatial diversity and, subsequently, the most aggressive opponent of the group has the opportunity to reproduce. This combination rewards both aggression explicitly and spatial diversity implicitly. Since estimating the interest value in real-time is an expensive procedure, aggression (via (3)) determines the interest value estimate that guides the search towards more interesting opponents.

The primary reason why OLL is successful is because it is based conceptually on an active player-opponent interaction. Hence, the more aggressive the opponents become through (3), the more challenging the game is for the player. Since, the player actively attempts to avoid them, it increases the spatial diversity of the opponents that are trying to follow him/her by uniformly covering the game environment. These behaviors collectively lead to the satisfaction of the challenge and the spatial diversity criteria in (1). Fig. 7 illustrates the dependencies between the T, S, $E\{H_n\}$ and I values over on-line learning games. In the specific experiment, we let a group of B *Ghosts* play against the ADV *PacMan* in the Normal stage and we record the aforementioned values every 100 games.

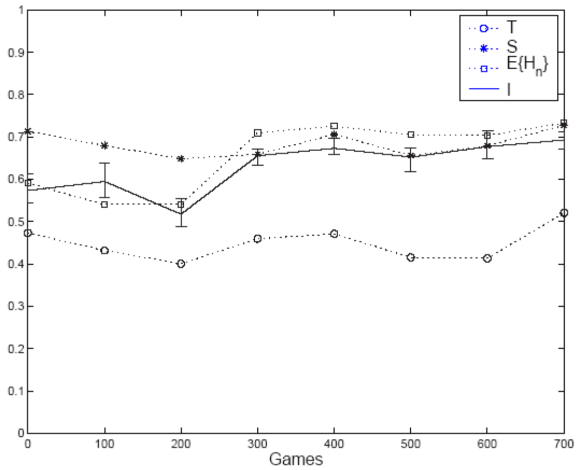


Fig. 7. Comparison figure of T, S, $E\{H_n\}$ and I. Initial opponent behavior: Blocking.

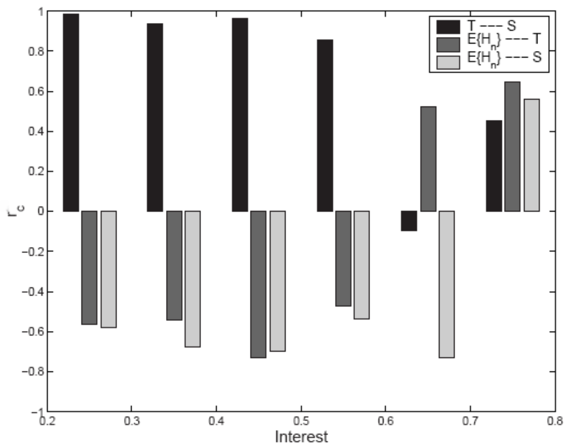


Fig. 8. Correlation coefficients of OLL generated T, S and $E\{H_n\}$ values over I value intervals.

As seen from Fig. 7, OLL initially increases the *Ghosts*' spatial diversity ($E\{H_n\}$) which furthermore produces more appropriate challenge for the player (T) through the player-opponent interaction. Ultimately, the game reaches its highest interest value when the player discovers new ways of playing that the opponents can counter (increase of behavior diversity — S).

Additional evidence of the OLL approach's behavior is presented in Fig. 8 where the correlation coefficients (r_c) between T, S and $E\{H_n\}$ values over the I value intervals are illustrated. A number of instances of these 128 values is obtained from three OLL experiments (B, A and H initial *Ghosts*) against the ADV *PacMan* in the Normal stage. According to Fig. 8, when $I < 0.6$, T and S are highly correlated and $E\{H_n\}$ is highly anticorrelated with both T and S. When $0.6 \leq I < 0.7$, T and S are slightly anticorrelated ($r_c = -0.0974$) and $E\{H_n\}$ is highly correlated and anticorrelated with T ($r_c = 0.5247$) and S ($r_c = -0.7296$) respectively. Finally, when $I \geq 0.7$, all three interest criteria are highly correlated. These correlation coefficients denote that high spatial diversity is likely to produce higher challenge (when $I \geq 0.6$) and furthermore higher *Ghosts*' behavior diversity when $I \geq 0.7$.

XV. Adaptability Experiments

On-line learning in Dead End revealed adaptive behavior to new — unknown during off-line training — playing strategies that lead to the improvement of the player's entertainment (see Fig. 6). Additional experiments are held here in the Pac-Man game in order to further demonstrate the mechanism's adaptability.

In order to test the OLL approach's ability to adapt to a changing environment (i.e. change of the *PacMan*'s strategy), the following experiment is proposed. Beginning from an initial behavior of high interest value I_{init} we apply the OLL mechanism against a specific *PacMan* type. During the on-line process we keep changing the type of player as soon as interesting games (i.e. $I \geq I_{init}$) are produced. The process stops when all three types of players have played the game. We repeat this process for all four stages examined; however, for space considerations, results presented here are obtained from experiments in the most difficult adaptability case-study, that is the Hard stage.

Since we have three types of players, the total number of different such experiments is 6 (all different player type sequences) for each stage. These experiments illustrate the overall picture of the approach's behavior against any sequence of *PacMan* types. As seen in Fig. 9, OLL is able to quickly recover form a sudden change in the player's strategy even in the most difficult stage and boost the game's interest value at high levels after sufficient games (i.e. 100 to 1500) have been played. The mechanism demonstrates a similar adaptive behavior for all 6 different sequences of *PacMan* players and all four stages which illustrates its independence of the sequence of the changing *PacMan* type and the stage complexity.

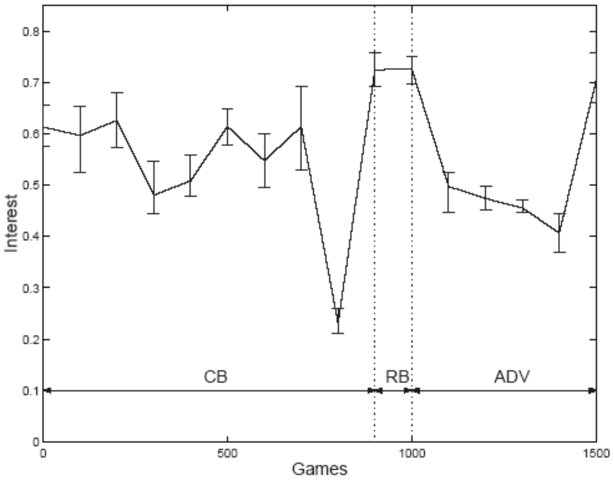
Results obtained from this experiment provide evidence for the approach's ability to adapt to new types of players as well as its efficiency in producing interesting games against human players.

XVI. Discussion

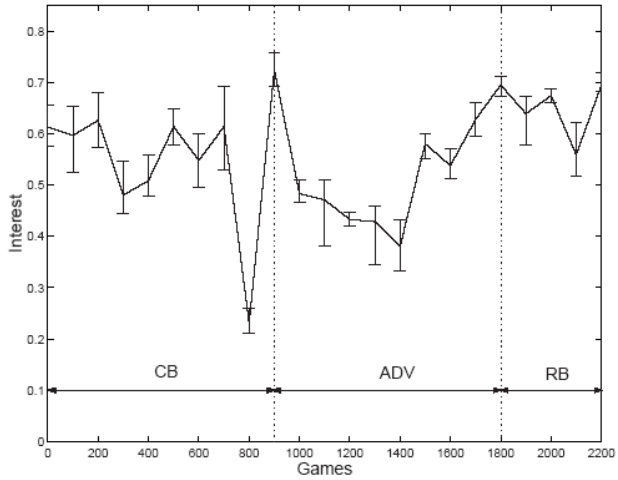
This section provides a discussion of the proposed methodology for obtaining predator/prey games of high entertainment value, by outlining a summary of its demonstrated generality over the dimensions of game variants, game complexity, player and initial behavior. Moreover, this section portrays the basic assumptions drawn for this work and solutions to potential drawbacks emerging when games scale up.

A. Predator/Prey Game Variants

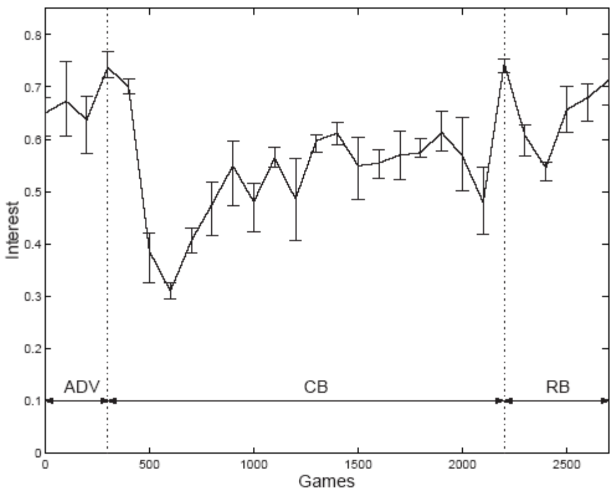
For the experiments presented here we used two predator/prey games differing in the characters' motion type, stage environment and player objectives. Interest reached high values and the OLL mechanism demonstrated robustness and adaptability when applied to both Pac-Man and Dead End games. However, no effective comparison between the two games' generated interest values can be derived and therefore



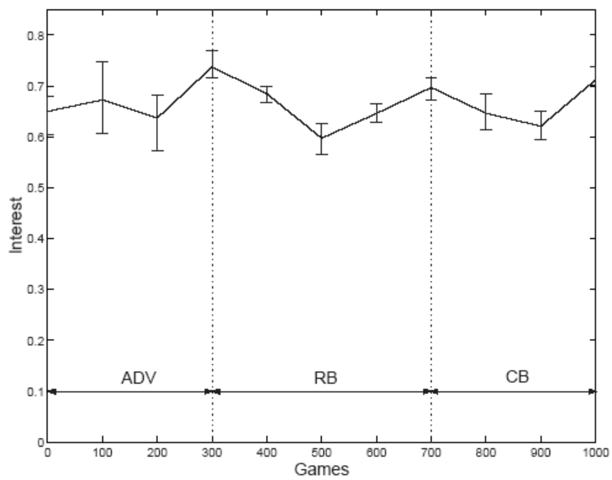
(a) CB-RB-ADV



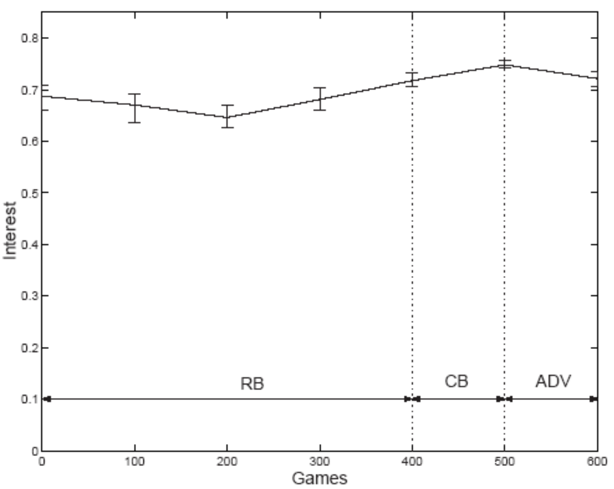
(b) CB-ADV-RB



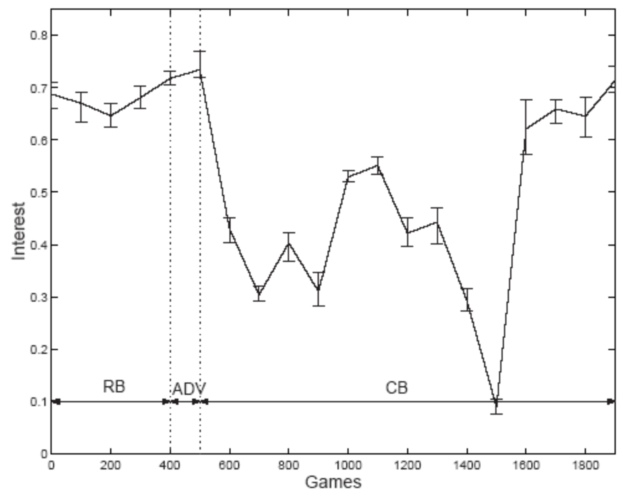
(c) ADV-CB-RB



(d) ADV-RB-CB



(e) RB-CB-ADV



(f) RB-ADV-CB

Fig. 9. Hard stage: On-line learning *Ghosts* playing against changing types of *PacMan*. Sub-figure captions indicate the playing *PacMan* sequence.

no answer can be given to which game is more interesting by design. Based on the games' main features and differences, we believe that these two test-beds cover a large portion of the properties met in the predator/prey computer game genre.

B. OLL Variants

Regarding the OLL approach variants used for the two games, experiments project a smooth but rather slow change of the interest value in Pac-Man whereas in Dead End a quite noisy (unstable) but relatively fast change is noticed. Moreover, the more complex the Dead End game environment (i.e. fewer *Dogs*) is, the more distinctive this instability becomes. The aforementioned dissimilarity in the interest value evolution is fully determined by (a) the pre-evaluation procedure and (b) the gaussian mutation operator that exist in the Pac-Man version of the OLL mechanism. According to the former, the probability of disruptive phenomena caused by unsuccessful mutations is minimized and, therefore, evolution is decelerated for the sake of smooth changes in emergent behaviors. According to the latter, the variance of the gaussian mutation is inversely proportional to the entropy of a group of *Ghosts*. Hence, the higher the *Ghosts*' cell visit entropy, the less disruptive the mutation process.

If we attempt to compare the two mechanisms used, the Pac-Man OLL variant appears as a more sophisticated algorithm that is designed to skip undesired opponent behaviors and erratic changes of the I value with the cost of convergence time. On the contrary, the Dead End OLL variant is a hill-climber that generates interesting games faster with the cost of instability.

C. Game Complexity & Topology

The OLL's ability to generate games of high interest value was tested over varying game complexity, which corresponds to the Easy, Normal and Hard stages for Pac-Man and to five, four and three *Dog* environments for Dead End. Results obtained from these experiments demonstrate the approach's generality since interesting games emerge independently of game complexity.

In addition to stages of different complexity, topologically different Pac-Man stages of equal complexity (Easy A and Easy B) were used as test-beds for the approach. Obtained interest values showed that the topology of the stage does not seem to hinder the OLL's adaptive features surfacing.

D. Player

As far as the playing strategy is concerned, three hand-crafted players have been designed for each game. Independently of playing strategy, the mechanism adapted to their playing style in order to boost the game's interest value. More specifically, in the Pac-Man game, the best OLL generated interest values were not significantly different regardless the player type.

The main assumption for both games is that players overall have a basic level of gaming skills in each game. In that sense, the computer guided Pac-Man players are models of some well-behaved, average-skill players based on similar motion patterns that do not leave much space for significant differences in their performance. In addition, humans who have tested Pac-Man validate this assumption since their generated interest values against the same opponent were not significantly different [2]. However, further studies using player modelling have displayed its positive impact on the generation of more interesting Pac-Man games over a longer game time scale [23].

According to the overall observed behavior of the I value it appears that, in addition to the opponent, the player may also play a significant role in its own entertainment. The player may determine the game's plot to a degree and this occurs due to its interaction with the opponents. In that sense, the interest value is affected by the player more in extreme scenarios such as unacceptably low (i.e. unable to control the player and sense the features of the game environment) or expert (i.e. unbeatable) gaming skills.

For instance, both EA and RM (in a lesser degree) types of *Cat* belong to the above-mentioned category of game-playing by following a trivial low-quality game strategy. Contrary to the PFB *Cat* and all *PacMan* types used, the aforementioned player types do not interact with their opponents, which furthermore, leads to a poor estimation of the game's I value. Thus, EA *Cat* appears to generate significantly more interesting games than the RM *Cat* and the PFB *Cat*. Note that the averages of the best interest values achieved from OLL experiments presented in Section XIII-A are 0.7641, 0.7240 and 0.7118 when playing against the EA, the RM and the PFB *Cat* respectively. Even though such playing strategies (e.g. EA and RM *Cat*) are rare among humans they constitute a limitation of the interest value estimation.

E. Initial Behavior

Five different behaviors emerged from off-line training procedures were selected as initial points in the search for more interesting games. For the Pac-Man game we categorized OLT emerged behaviors into blocking, aggressive and hybrid whereas for the Dead End game the OLT behaviors obtained were characterized as either aggressive or defensive. Given these diverse initial behaviors, the OLL mechanism exhibited high robustness and fast adaptability in increasing the game's interest value. Moreover, results showed that convergence time of highly interesting games is dependent on the initial interest value.

XVII. Conclusions

Predator strategies in predator/prey computer games are still nowadays based on simple rules which generate complex opponent behaviors [24]. However, by the time the player gains more experience and playing skills the game becomes rather predictable and, therefore, somewhat uninteresting. A computer game becomes interesting primarily when there is an on-line interaction between the player and his opponents who demonstrate adaptive behaviors [25].

Given some criteria for defining interest in predator/prey games we applied a methodology for explicitly measuring a human-verified entertainment value in real-time in such games. We saw that by using the proposed on-line learning mechanism in two dissimilar games, maximization of the individual objective function (see (3)) leads to maximization of the game's interest value. Apart from being robust, the proposed mechanism demonstrates high and fast adaptability to new types of player (i.e. playing strategies). Results obtained from these experiments demonstrate the approach's generality since interesting games emerge independently of game concept, stage complexity, initial opponent behavior and player type.

Based on the experiments presented in this paper, the neuro-evolution methodology proposed reveals features that overcome the three main challenges of learning in real-time in games (see Section IV). More specifically, interesting games were generated using a single CPU (real-time performance); the probability of generating unwanted behaviors is minimized (Pac-Man version of OLL) when the game generates high levels of interestingness; and the opponents' adaptability time reaches the levels of tens of games when the initial opponent behavior is highly interesting.

A. Extensibility

Experiments over variants of predator/prey games of different complexity and their features have demonstrated the methodology's robustness throughout this paper. Here, we discuss the potential of the methodology in other genres of multi-opponent games where the complication of the opponents' tasks may differ. More specifically, we analyze the extensibility of the interest metric proposed, the on-line evolutionary learning mechanism and the neuro-controller used.

1) Interest Metric: As already mentioned in Section V, the criteria of challenge and behavior's diversity may be effectively applied for measuring the real-time entertainment value of any genre of games. Spatial diversity may in a sense also contribute to the interest value of specific genres (e.g. team-sport, real-time strategy and first-person shooter games). As long as game developers can determine and extract the features of the opponent behavior that generate excitement for the player, a mathematical formula can be designed in order to collectively represent them.

2) Learning Methodology: The proposed on-line evolutionary learning method may also be successfully applied to any game during active real-time player-opponent interactions. Extracted features of this interaction may be used in order to estimate the fitness of the involved opponents according to their tasks. The replacement of the worst-fit opponent(s) method may be applied in frequent game periods to enhance the group's fitness. See also Stanley et al. [14] for a successful application of this method in the NERO game.

Artificial evolution can explore complex search spaces efficiently and when combined with neural networks it can demonstrate fast adaptability to dynamic and changing environments. Therefore neuro-evolution is recommended for learning in real-time. However, convergence time and unpredictability of the emergent behaviors constitute the disadvantages of the methodology which can be dealt with by careful design of the learning mechanism. Player modeling techniques are able to decrease convergence time down to realistic periods of time (i.e. tens of games) and furthermore proliferate the efficiency and justifiability of learning in real-time [23].

3) Controller: Artificial neural networks serve successfully the adaptability requirements for predator/prey reactive games in real-time. However, as the complexity of the opponents' tasks increases there might be a need for more sophisticated structures of distributed representation. Memory of previous behaviors learned through the player-opponent interaction may very well be essential when a combination of various tasks is required. Recurrent neural networks or augmented neural network topologies with hidden states [16] may be more appropriate when the opponents' tasks proliferate. Moreover, a hierarchy of neuro-controllers that serve different opponent tasks could also provide the on-line learning mechanism with more flexibility and faster adaptability.

References

[1] G. N. Yannakakis and J. Hallam, "Evolving Opponents for Interesting Interactive Computer Games," in *From Animals to Animats 8: Proceedings of the 8^{th} International Conference on Simulation of Adaptive Behavior (SAB-04)*, S. Schaal, A. Ijspeert, A. Billard, S. Vijayakumar, J. Hallam, and J.-A. Meyer, Eds. Santa Monica, LA, CA: The MIT Press, July 2004, pp. 499–508.

[2] ——, "Towards Optimizing Entertainment in Computer Games," June 2005, submitted to Applied Artificial Intelligence.

[3] ——, "A generic approach for generating interesting interactive pac-man opponents," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, G. Kendall and S. Lucas, Eds., Essex University, Colchester, UK, 4–6 April 2005, pp. 94–101.

[4] ——, "Interactive Opponents Generate Interesting Games," in *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, Microsoft Campus, Reading, UK, November 2004, pp. 240–247.

[5] J. E. Laird and M. van Lent, "Human-level AI's killer application: Interactive computer games," in *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI)*, 2000, pp. 1171–1178.

[6] C. Thurau, C. Bauckhage, and G. Sagerer, "Learning human-like Movement Behavior for Computer Games," in *From Animals to Animats 8: Proceedings of the 8^{th} International Conference on Simulation of Adaptive Behavior (SAB-04)*, S. Schaal, A. Ijspeert, A. Billard, S. Vijayakumar, J. Hallam, and J.-A. Meyer, Eds. Santa Monica, LA, CA: The MIT Press, July 2004, pp. 315–323.

[7] A. Nareyek, "Intelligent agents for computer games," in *Computers and Games, Second International Conference, CG 2002*, T. Marsland and I. Frank, Eds., 2002, pp. 414–422.

[8] S. Lucas, "Evolving a neural network location evaluator to play Ms. Pac-Man." in *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, G. Kendall and S. Lucas, Eds., Essex University, Colchester, UK, 4–6 April 2005, pp. 203–210.

[9] M. Freed, T. Bear, H. Goldman, G. Hyatt, P. Reber, A. Sylvan, and J. Tauber, "Towards more human-like computer opponents," in *Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, 2000, pp. 22–26.

[10] N. A. Taatgen, M. van Oploo, J. Braaksma, and J. Niemantsverdriet, "How to construct a believable opponent using cognitive modeling in the game of set," in *Proceedings of the fifth international conference on cognitive modeling*, 2003, pp. 201–206.

[11] H. Iida, N. Takeshita, and J. Yoshimura, "A metric for entertainment of boardgames: its implication for evolution of chess variants," in *IWEC2002 Proceedings*, R. Nakatsu and J. Hoshino, Eds. Kluwer, 2003, pp. 65–72.

[12] T. W. Malone, "What makes computer games fun?" *Byte*, vol. 6, pp. 258–277, 1981.

[13] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York: Harper & Row, 1990.

[14] K. Stanley, B. Bryant, and R. Miikkulainen, "Real-time evolution in the NERO video game," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, G. Kendall and S. Lucas, Eds., Essex University, Colchester, UK, 4–6 April 2005, pp. 182–189.

[15] G. N. Yannakakis, J. Levine, and J. Hallam, "An Evolutionary Approach for Interactive Computer Games," in *Proceedings of the Congress on Evolutionary Computation (CEC-04)*, June 2004, pp. 986–993.
[16] K. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.
[17] S. Woodcock, "Game AI: The State of the Industry 2000-2001: It's not Just Art, It's Engineering," August 2001.
[18] J. A. Meyer and A. Guillot, "From SAB90 to SAB94: Four years of animat research," in *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior (SAB-92)*, 1994.
[19] A. J. Champandard, *AI Game Development*. New Riders Publishing, 2004.
[20] G. N. Yannakakis, J. Levine, J. Hallam, and M. Papageorgiou, "Performance, robustness and effort cost comparison of machine learning mechanisms in *FlatLand*," in *Proceedings of the 11th Mediterranean Conference on Control and Automation MED'03*. IEEE, June 2003.
[21] J. K. Park, "Emerging complex behaviors in dynamic multi-agent computer games," M.Sc. Thesis, University of Edinburgh, 2003.
[22] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. for Robotics Research*, vol. 5, no. 1, pp. 90–99, 1986.
[23] G. N. Yannakakis and M. Maragoudakis, "Player modeling impact on player's entertainment in computer games," in *Proceedings of the 10^{th} International Conference on User Modeling; Lecture Notes in Computer Science*, vol. 3538. Edinburgh: Springer-Verlag, 24–30 July 2005, pp. 74–78.
[24] Wikipedia, the Free Encyclopedia, "Pac-Man," June 2005. [Online]. Available: http://en.wikipedia.org/wiki/Pac-man
[25] G. N. Yannakakis and J. Hallam, "A scheme for creating digital entertainment with substance," in *Proceedings of the Workshop on Reasoning, Representation, and Learning in Computer Games, 19th International Joint Conference on Artificial Intelligence (IJCAI)*, August 2005, pp. 119–124.